

卒 業 論 文

自動車運搬船における貨物積載プランニング の車両配置問題に対する構築型解法

101810106 黒須諒

名古屋大学情報学部
自然情報学科数理情報系

2022年2月

自動車運搬船における貨物積載プランニングの 車両配置計画に対する構築型解法

101810106 黒須諒

概要

複数の港で自動車を積み、複数の港で自動車を降ろす自動車運搬船について考える。自動車運搬船に積む自動車の集合が与えられてから実際に自動車が船に積まれるまでに、席割とシミュレーションと呼ばれる二種類の作業が行われる。席割作業では、船の階層内を一定間隔の大きさに区切ったホールドと呼ばれるスペースの各々に、与えられた積載自動車リストの自動車を何台割り当てられるかを考える。シミュレーション作業では席割作業でホールド毎に割り当てられた自動車を、自動車の向きや空きスペース、作業効率などを考慮して車一台一台の配置場所を決定する。車両配置図を出力することで、現場の作業員が配置図に従い、効率よく駐車作業を行うことができる。本研究では、シミュレーション作業の自動化を目的とし、短時間かつ実用的なシミュレーション結果を出力することを目指す。

シミュレーション作業では車の幅と長さを各辺とした長方形に近似することで、長方形詰込み問題として定式化することが出来るが、貨物である自動車は配置位置まで自走するという事を考えなければいけない。そのため、積み降ろしのタイミングでは配置場所までの搬入・搬出経路の確保と、駐車時に要する局所的スペースの確保が課題となる。

本研究では二段階の構築法と局所探索を提案する。一段階目では、積み地、揚げ地が同じ車を1つのグループとし、デッキ内におけるグループの大まかな配置場所を決定する。グループに属する車の総面積を求め、面積制約をもち、長方形の高さと幅を変数とする soft-rectangle パッキング問題として定式化する。各グループの積み地、揚げ地に関する順序対を用意し sequence-pair を用いた制約を追加することで、搬入搬出時の移動経路を確保した解が得られる。この計算には、商用の整数計画ソルバー (Gurobi Optimizer ver. 9.5) を使用した。

二段階目では、第一段階で決められた領域内で車両一台ずつの詳細な配置場所を決定する。配置予定の車を、駐車に必要なスペースを加えたレクトリニア図形に近似し、レクトリニア図形の詰込み問題として定式化する。車の向きを考慮した bottom-left 法を用いて実装した。以上の構築が終わった後、より多くの車を詰め込むために局所探索を行った。

以上の提案手法により得られた車両配置図と実際に現場で使われている車両配置図とを比較し、解の評価を行った。その結果、多くの問題例において、搬入搬出時の経路確保と駐車時の局所的スペース確保ができていることを確認した。また、構築のみの実装により詰め込むことのできた車の台数と、局所探索を含めた実装で詰め込むことのできた車の台数を比較した結果、多くの問題例で局所探索を含めた実装の方がより多くの車を詰め込めることを確認した。

目次

第 1 章	はじめに	1
第 2 章	問題定義	2
2.1	用語定義	2
2.2	入力情報	3
2.3	出力	3
第 3 章	定式化	4
3.1	記号の定義	4
3.2	一般論 (仮タイトル)	4
3.2.1	長方形詰込み問題	4
3.2.2	Soft-rectangle パッキング問題	5
3.2.3	Sequence-pair	5
3.2.4	No-fit polygon	6
3.3	第一段階	6
3.3.1	制約	6
3.3.2	目的関数	7
3.3.3	定式化	7
3.4	第二段階	8
3.4.1	制約	8
3.4.2	目的関数	8
3.4.3	定式化	8
第 4 章	提案手法	9
4.1	第一段階	9
4.2	第二段階	9
4.3	局所探索法	9
第 5 章	計算実験	11
5.1	計算環境	11
5.2	第一段階の求解結果	11
5.3	第二段階の求解結果	12
5.4	局所探索法による解の改善	12
第 6 章	まとめと今後の研究計画	13
	参考文献	15

第1章 はじめに

複数の港で自動車を積み、複数の港で自動車を降ろす自動車運搬船について考える。一般的に自動車運搬船は、自動車を船の一定間隔で区切られたホールドと呼ばれるスペースにどの自動車は何台割り当てるかを決定する席割と呼ばれる工程を経て、その後席割作業で割り当てられた自動車に対して向きや場所を考慮して一台ずつ船内の領域に配置するシミュレーションと呼ばれる作業を行う。貨物である自動車は積載位置まで自走するため、積み降ろしのタイミングでの搬入・搬出経路を確保する必要がある。また、航海中の船体バランス及び荷役安全性を考慮する必要がある。現状、自動車を輸送する会社はこの作業を人手で行なっていることが多く、席割作業に3時間、シミュレーション作業に4時間かかると言われている [1]。プランナーごとの経験値や技量によって積み付け計画の品質に個人差が生じ、急な状況の変化による積み付け計画の変更に業務負荷が生じるなどの問題が存在している。名古屋大学の竹田が積み付け計画自動化の第一歩として、席割作業の自動化に関する研究を進めている [2]。本研究では積み付け計画自動化の次の段階としてシミュレーション作業自動化を目的とし、短時間で実用的なシミュレーション結果を出力することを目指す。

本研究で扱うシミュレーション作業の概要について述べる。入力として、各階層の各ホールドにどの種類の車を何台詰め込むかという情報が与えられる。この情報をもとにシミュレーション作業を行う。シミュレーション作業は、車の幅と長さを各辺とする長方形に近似することで長方形詰め込み問題として定式化できるが、貨物である自動車は積載位置まで自走するということを考える必要がある。そのため、積み降ろしのタイミングでの搬入・搬出経路の確保と、駐車時に要する局所的スペースの確保が課題となる。

本研究では、二段階に分けた構築法と局所探索を提案する。一段階目では、各ホールド内に詰め込む車を、積み地や揚げ地の情報をもとにグループ分けし、グループごとにデッキ内での大まかな配置場所を決める。二段階目では、グループ内における車両一台ずつの詳細な配置場所を決定する。以上の二段階による構築法が終わった後、簡単な局所探索を行い解を改善する。

本稿では、第2章で問題に対する詳細な設定や、本研究で扱う自動車運輸航海における専門用語について定義する。第3章では、問題の定式化と定式化に必要な変数や定数の定義を行う。第4章と第5章では、提案手法の説明と解の考察を行う。

第2章 問題定義

シミュレーション作業は各階層ごとに行う。各階は4つのホールドと呼ばれる一定の広さで区切られたスペースが存在し、各ホールドに詰め込む車の情報は決まっている。本稿では複数種類のある階層の中で、4つのホールドを持つ単純な階層のデータを用いて行う。

2.1 用語定義

本研究で扱う船の航海等に関する専門用語の定義をする。用語や記号は関連研究 [2] に従う。

- 席割
注文一つ一つを船のホールドに割り当てる作業。
- シミュレーション
席割で決まった自動車を一台ずつホールド内の領域に貼り付ける作業。
- プランナー
席割やシミュレーションを考える作業。
- オペレーター
港で自動車をホールドまで運転して積み降ろしをする作業。
- デッキ
船の内部の階層。
- ホールド
各デッキ内を一定間隔の領域で仕切られた空間。
本研究では、各デッキは4つのホールドを持つ。
- セグメント
隣接するいくつかのホールドをまとめた空間。
本研究では、ホールド 1,2 とホールド 3,4 をまとめたものをそれぞれセグメント 1, セグメント 2 と呼ぶ。
- ランプ
上下のデッキに移動するために各デッキの特定ホールドについているスロープ。
- 積み地
注文における自動車を積む港。
- 揚げ地
注文における自動車を降ろす港。

2.2 入力情報

本研究で扱う 2 種類の情報について述べる.

- 車体情報
車の数, 種類, 大きさ, 積み地, 揚げ地, 指定のホールド, ハンドル向き.
本研究では実際のプランで使われているデータを利用した.
- 船体情報
船体の大きさ, 各階のランプの位置, 船内の障害物の位置や大きさ.
ランプも障害物の一つとして計算している.
本研究では実際の図面データを元に手入力で行った.

入力情報のうち車体情報の例を表 2.1 に示す.

表 2.1: 車体情報の例

番号	車種	積み地	揚げ地	数量	ホールド	幅	長さ	ハンドル
1	Prius	A 港	E 港	100	1	200	500	r
2	Prius	A 港	E 港	100	2	200	500	r
3	Aqua	B 港	F 港	30	2	250	450	l
4	Carolla	C 港	G 港	60	3	180	480	l
5	Lexus	C 港	G 港	40	3	230	510	l
6	Lexus	D 港	H 港	100	4	230	420	r

2.3 出力

詰め込むことのできた車の台数と配置図を出力する.
配置図の例を図 2.1 に示す.

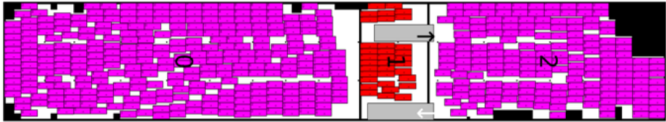


図 2.1: 車両配置図の例

第3章 定式化

この章では1節で本研究における記号の定義や問題の定式化について説明し, 2節では, 本研究に関連する詰め込み問題の説明をする. その後に, 本問題に関する独自の制約や目的関数を説明し, 定式化する.

3.1 記号の定義

記号	記号の説明
x_i	長方形 i の左下の x 座標
y_i	長方形 i の左下の y 座標
w_i	長方形 i の幅
w_i^l	長方形 i の幅の下界
w_i^u	長方形 i の幅の上界
h_i	長方形 i の高さ
h_i^l	長方形 i の高さの下界
h_i^u	長方形 i の高さの上界
W	母材の横幅
H	母材の高さ
S_i	長方形 i の面積
I	詰め込む長方形の集合
\bar{I}	既配置の長方形の集合 $\bar{I} \subseteq I$
J	障害物の集合

3.2 一般論 (仮タイトル)

3.2.1 長方形詰め込み問題

長方形詰め込み問題とは, 母材の幅 W と高さ H , 長方形集合 I に対し, 幅 w_i , 高さ h_i が入力として与えられ, 母材からはみ出ることなくまた, どの二つの長方形も互いに重ならないように詰め込む問題である.

以上の条件を定式化すると, 次のように書くことができる.

条件 1: 長方形 $i \in I$ は母材上に配置される.

$$0 \leq x_i \leq W - w_i \quad (i \in I), \quad (3.1)$$

$$0 \leq y_i \leq H - h_i \quad (i \in I). \quad (3.2)$$

条件 2: 長方形 $i, j \in I$ は互いに重ならない.

この条件は, 次の 4 つの不等式のうち一つ以上が成立しなければならない.

$$x_i + w_i \leq x_j \quad (i \in I), \quad (3.3)$$

$$x_j + w_j \leq x_i \quad (i \in I), \quad (3.4)$$

$$y_i + h_i \leq y_j \quad (i \in I), \quad (3.5)$$

$$y_j + h_j \leq y_i \quad (i \in I). \quad (3.6)$$

3.2.2 Soft-rectangle パッキング問題

長方形詰込み問題の一種で, 詰め込む長方形の幅と高さを可変とした soft-rectangle (可変形状長方形) を扱う [4]. 長方形 $i \in I$ の座標 x_i, y_i だけでなく幅 w_i と高さ h_i も変数とする.

通常, 長方形 $i \in I$ の面積に関する次のような等式制約または不等式制約が存在する:

$$w_i * h_i = S_i, \quad (3.7)$$

$$w_i * h_i \geq S_i. \quad (3.8)$$

$$(3.9)$$

また, 長方形 $i \in I$ の幅と高さの上界, 下界に関する以下の制約も存在する:

$$w_i^l \leq w_i \leq w_i^u, \quad (3.10)$$

$$h_i^l \leq h_i \leq h_i^u. \quad (3.11)$$

3.2.3 Sequence-pair

Sequence-pair とは, 二つの順列による長方形の配置表現である [5]. 長方形同士の相対位置関係を長方形名の順列対により表すことができる. 例を図 3.1 に示す. 制約の詳細は 3.3 節で説明する. Sequence-pair はどのような配置表現でも一意の順列対に表現でき, 逆に任意の順列対から一意に配置表現を求めることができる.

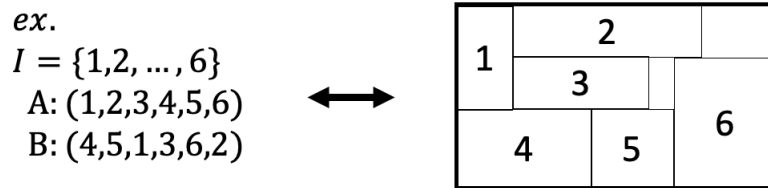


図 3.1: sequence-pair による長方形配置表現

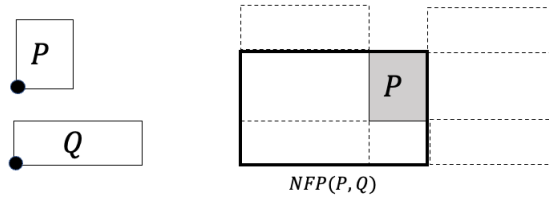


図 3.2: $NFP(P, Q)$

3.2.4 No-fit polygon

No-fit polygon (NFP) とは、平面上で多角形の重なりを判定する方法である [6][7]. 多角形 P と Q が与えられ、 Q の配置が固定されているとする. この時、多角形 P が Q と重なりを持つ領域を $NFP(P, Q)$ と書き、図 3.2 に示す. 図 3.2 の太線が NFP の境界を表し、この NFP 内に P を配置すると Q と重なることを意味する. 母材内の既配置の長方形全てに対し、長方形 i の NFP を作成すると、長方形 i の配置可能領域はどの NFP の内部にも含まれない領域である.

3.3 第一段階

第一段階では、積み地・揚げ地が同じ車を一つのグループとし、各グループのデッキ内における大まかな配置場所を決定する. グループ内の車の総面積を計算し、soft-rectangle パッキング問題として定式化する. このパッキングはセグメントごとに行う.

3.3.1 制約

(i) ランプから配置場所までの移動経路確保に関する制約

プランナーへのヒアリングの結果から、先に積む貨物ほど入口から見て奥に積み、後から積む貨物ほど手前に積むことが望ましいと分かった. 同様に、先に降ろす貨物ほど入口から見て手前に積み、後から下ろす荷物ほど奥に積むこともまた望まれる. 本研究では、Sequence-pair を用いてグループの相対位置関係に関する制約を表現する. 各グループを積み地の昇順に並べた順列を σ^+ とする. 積み地が同じグループが存在した場合、揚げ地の降順に従う. 同様に、揚げ地の昇順に並べた順列を σ^- とする. 揚げ地が同じグループが存在する場合、積み地の降順に従う. グループの集合を I とし、グループ $i, j \in I$ に関して、順序対 (σ^+, σ^-) における i, j の順序関係をもとに、以下のどちらかの制約を追加する:

$$\begin{cases} x_i + w_i \leq x_j & \sigma^+(\dots, i, \dots, j, \dots) \cap \sigma^-(\dots, i, \dots, j, \dots) \\ y_i + h_i \leq y_j & \sigma^+(\dots, i, \dots, j, \dots) \cap \sigma^-(\dots, j, \dots, i, \dots) \end{cases} \quad (3.12)$$

(ii) 各グループの面積に関する制約

次の段階で各グループの車を詰め込むためには、グループの面積はグループ内に属する車の総面積よりも大きくなければいけない. 以上の制約は、グループの集合を I とし、グループ $i \in I$ の車の総面積を S_i とすると、以下のように書ける:

$$w_i * h_i \geq S_i, \quad (3.13)$$

一つのグループの領域が大きくなりすぎると、デッドスペースが生まれたり、他のグループの領域で詰め込めない車が出てしまう可能性があるため、パラメータ変数 a ($a \geq 1$) を用意して、以下の

ように上から抑える.

$$w_i * h_i \leq a * S_i. \quad (3.14)$$

(iii) 各グループの幅と高さに関する制約

各グループ I の幅と高さは, グループ内に属する車 $i \in I$ の幅と長さの最大値をもとに以下の下限を設定する:

$$w_I \geq \max_{i \in I} w_i, \quad (3.15)$$

$$h_I \geq \max_{i \in I} h_i. \quad (3.16)$$

3.3.2 目的関数

目的関数として, デッキ内のより多くの領域を配置可能場所として使えることが望ましいので, グループの面積和の最大化を目指す. また1つのグループが大きくなりすぎると, デッドスペースが生まれたり, 他のグループの領域で詰め込めない車が出てしまう可能性があるので, バランス良く大きくすることを目指す. 面積制約 3.14 で用いたパラメータ a を用いて, 以下のように書くことができる:

$$\text{maximize } \sum_{i \in I} w_i * h_i - a. \quad (3.17)$$

3.3.3 定式化

パッキングするグループの集合を I とし, セグメントの幅と高さをそれぞれ W, H とし, 以上をまとめると以下ようになる.

$$\text{maximize } \sum_{i \in I} w_i - h_i - a \quad (3.18)$$

$$\text{subject to } 0 \leq x_i \leq W - w_i \quad (i \in I) \quad (3.19)$$

$$0 \leq y_i \leq H - h_i \quad (i \in I) \quad (3.20)$$

$$0 \leq w_i \leq \max_{car \in i} w_{car} \quad (i \in I) \quad (3.21)$$

$$0 \leq h_i \leq \max_{car \in i} h_{car} \quad (i \in I) \quad (3.22)$$

$$S_i \leq w_i * h_i \leq a * S_i \quad (i \in I) \quad (3.23)$$

$$\begin{cases} x_i + w_i \leq x_j & \sigma^+(\dots, i, \dots, j, \dots) \cap \sigma^-(\dots, i, \dots, j, \dots) \\ y_i + h_i \leq y_j & \sigma^+(\dots, i, \dots, j, \dots) \cap \sigma^-(\dots, j, \dots, i, \dots) \end{cases} \quad (3.24)$$

3.4 第二段階

第二段階では、第一段階で決められた領域をはみ出ることがないように、車一台ずつの配置場所を決定する。車を長方形に近似することで、長方形詰込み問題として定式化する。

3.4.1 制約

(i) 駐車に必要な局所的なスペースに関する制約

駐車は原則バック駐車で行うため、配置位置周辺に局所的なスペースが必要となる。例えば、配置位置のすぐ前方に柱等の障害物があると、配置図上では配置可能に見えても、実際には障害物が邪魔で駐車できない可能性がある。この点を考慮するため、予め駐車に必要なスペースを定義する。プランナーへのヒアリングの結果から、駐車にはおおよそ車体長さの半分に相当するスペースが必要で、このスペース上に障害物がなければ良いということが分かった。車 i の大きさと駐車に必要なスペースを合わせたレクトリニア図形を $f(i)$ で定義する。障害物の集合を J 、既配置の車の集合を \bar{I} とすると、車 i の配置不可能な領域は NFP の表現を用いて以下のように書くことができる。

$$\sum_{j \in J} NFP(f(i), j) + \sum_{\bar{i} \in \bar{I}} NFP(f(i), \bar{i}) \quad (3.25)$$

(ii) 配置間隔に関する制約

配置される車の間隔は他の車両や障害物と前後方向に 30 cm、左右方向に 10 cm 空いている必要がある。また、オペレーターの乗り降りのため、左右の内ハンドル側に 40 cm 空いている必要がある。先ほどの制約同様、配置予定の車を上記の単位だけ大きくして NFP を計算する。車両 i を上下方向に 30 cm ずつ、左右方向に 10 cm ずつ、ハンドル側を 40 cm 大きくした長方形を $g(i)$ とすると、車両 i の配置不可能な領域は以下のように表される。

$$\sum_{j \in J} NFP(g(i), j) + \sum_{\bar{i} \in \bar{I}} NFP(g(i), \bar{i}) \quad (3.26)$$

車両 i が配置可能な領域は、(3.25)、(3.26) 以外の領域である。

3.4.2 目的関数

第二段階では、目的関数の設定は行わず解の構築のみを行う。

3.4.3 定式化

まとめると、グループ I 内の車 i をパッキングする時の制約は、以下ようになる。

$$\text{subject to} \quad 0 \leq x_i \leq W_I - w_i \quad (i \in I) \quad (3.27)$$

$$0 \leq y_i \leq H_I - h_i \quad (i \in I) \quad (3.28)$$

$$x_i + w_i \leq x_j \quad \text{or} \quad x_j + w_j \leq x_i \quad \text{or} \quad y_i + h_i \leq y_j \quad \text{or} \quad y_j + h_j \leq y_i \quad (i, j \in I) \quad (3.29)$$

$$(x_i, y_i) \neq \sum_{j \in J} NFP(f(i) + g(i), j) + \sum_{\bar{i} \in \bar{I}} NFP(f(i) + g(i), \bar{i}) \quad (i \in I) \quad (3.30)$$

第4章 提案手法

本章では, 第3章で定式化した二段階の問題についてそれぞれの問題に対する提案手法を説明する.

4.1 第一段階

求解には整数計画ソルバー (Gurobi Optimizer ver. 9.5) を用いた. 第二段階では, この時の出力結果を利用する.

4.2 第二段階

bottom-left 法

bottom-left 法 (BL 法) とは長方形詰込み問題に対するアルゴリズムの一つで, 配置可能な領域の内, 最も下に, 同じ高さの候補があれば, 最も左に配置する手法である [7]. 面積の大きい順にパッキングするといったように, 詰め込む順番を工夫することで有用な解が得られるとされている.

本研究では, 車の駐車向きを考慮し, 搬出時ランプに向かって前向きに進出できるように BL 法を修正し, 詰め込みを行った. 各グループ内で, 全長の大きいもの, その中でも幅の大きいものを先に, そして同じ車種が連続するような順番で詰め込みを行った. グループ間の詰め込みの順番としては, 積み地の早い順, 積み地が同じ場合, 揚げ地の遅い順に詰め込みを行った.

グループ内の全ての車が詰め終わるか, 詰め込み途中で, グループ内のどの車も詰め込むことができない場合, 計算を終了し, 次のグループ内の車の詰め込みを開始した.

4.3 局所探索法

局所探索法

ある解 x に修正を加えることを近傍操作という [8]. 近傍操作により得られる解の集合 $N(x)$ を近傍と呼ぶ. 局所探索法は, 適当な初期解から始め, 現在の解 x よりも良い解 x' が近傍 $N(x)$ 内に存在すれば, $x := x'$ と置き換える操作を可能な限り繰り返す方法である.

本研究では, 二段階により得られた初期解に近傍操作を加え改善を行う手法を提案する. 評価関数としては, 詰め込むことのできた車の台数を用いる.

近傍操作では, 第一段階で決めたグループの大きさを変更する. 隣接する2つのグループ i, j を比較し, グループ i では全ての車が詰め終わり, グループ j では詰め込むことのできなかった車が存在するとする. このとき, グループ i の面積を少し小さくし, グループ j の面積を少し大きくする.

ここで変化させる面積はグループ j で詰め込むことのできなかった車の総面積分とする．以上の近傍操作を加えたのち，再度，第二段階の手法を用いて車を詰め込み，最終的に詰め込むことのできた車の台数を元の解と比較する．

第5章 計算実験

5.1 計算環境

実験に用いるプログラムは Python を用いて実装し, 計算機は MacBookAir2020 (CPU: 8 コア Apple M1 chip, メモリ: 8 GB LPDDR4) を用いて行った.

5.2 第一段階の求解結果

整数計画ソルバー (Gurobi Optimizer ver. 9.5) を用いて計算した結果, 全ての問題例で最適解が得られた. 出力の例を図 5.1 に示す. 長方形内の番号はグループの番号を表す. 灰色の長方形は, ランプを表す.

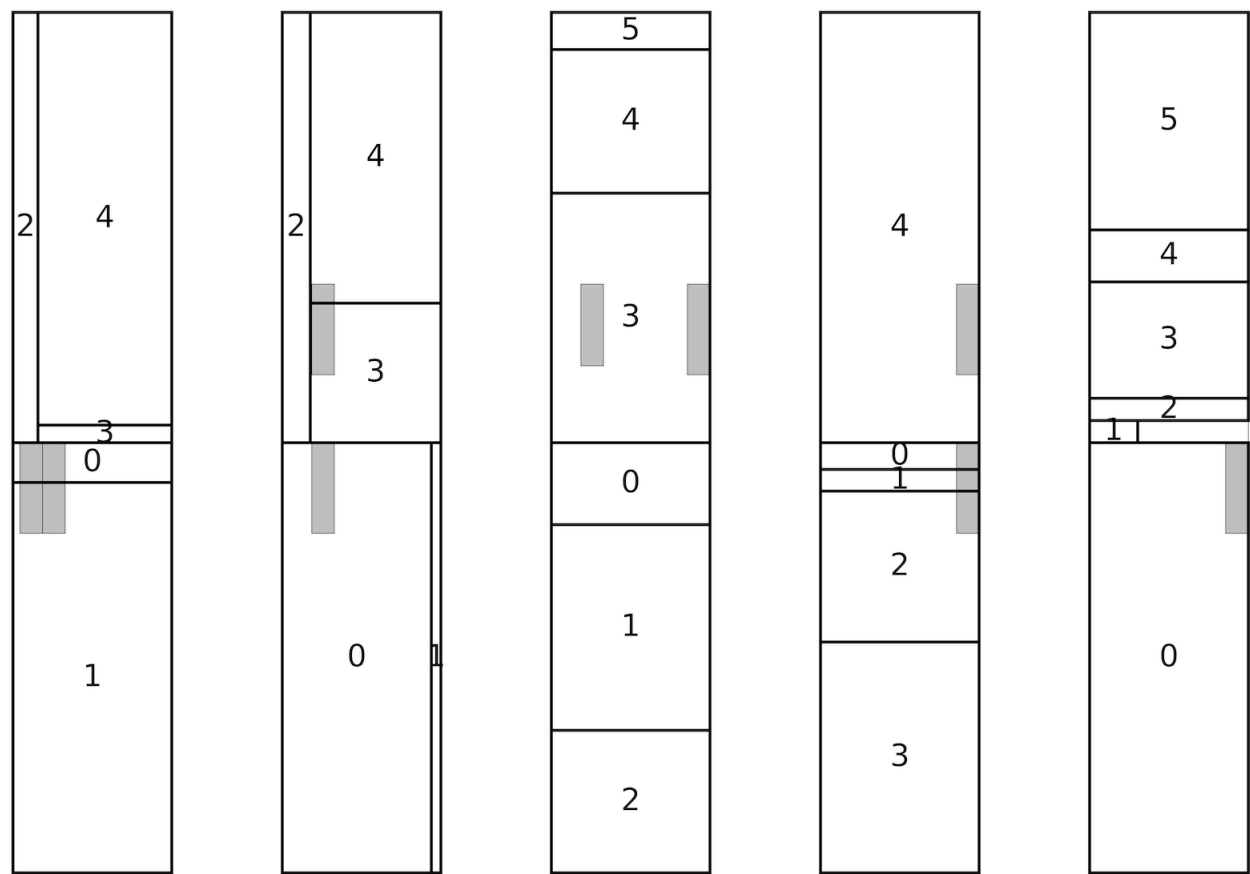


図 5.1: 第一段階の出力の例

5.3 第二段階の求解結果

出力される配置図の例を図 5.2 に示す。各長方形が車を表し、色は積み地の種類を表す。

5.4 局所探索法による解の改善

以下の表 5.1 に局所探索法を加えることで詰め込むことのできた解の改善結果を示す。表の台数は詰め込むことのできた車の数を表す。

表 5.1: 近傍操作による解の改善

問題例		初期解		局所探索	
問題番号	台数	台数	計算時間	台数	計算時間
1	773	753	142	765	447
2	604	534	274	572	618
3	594	554	271	560	583
4	481	459	233	481	414
5	276	593	276	592	395

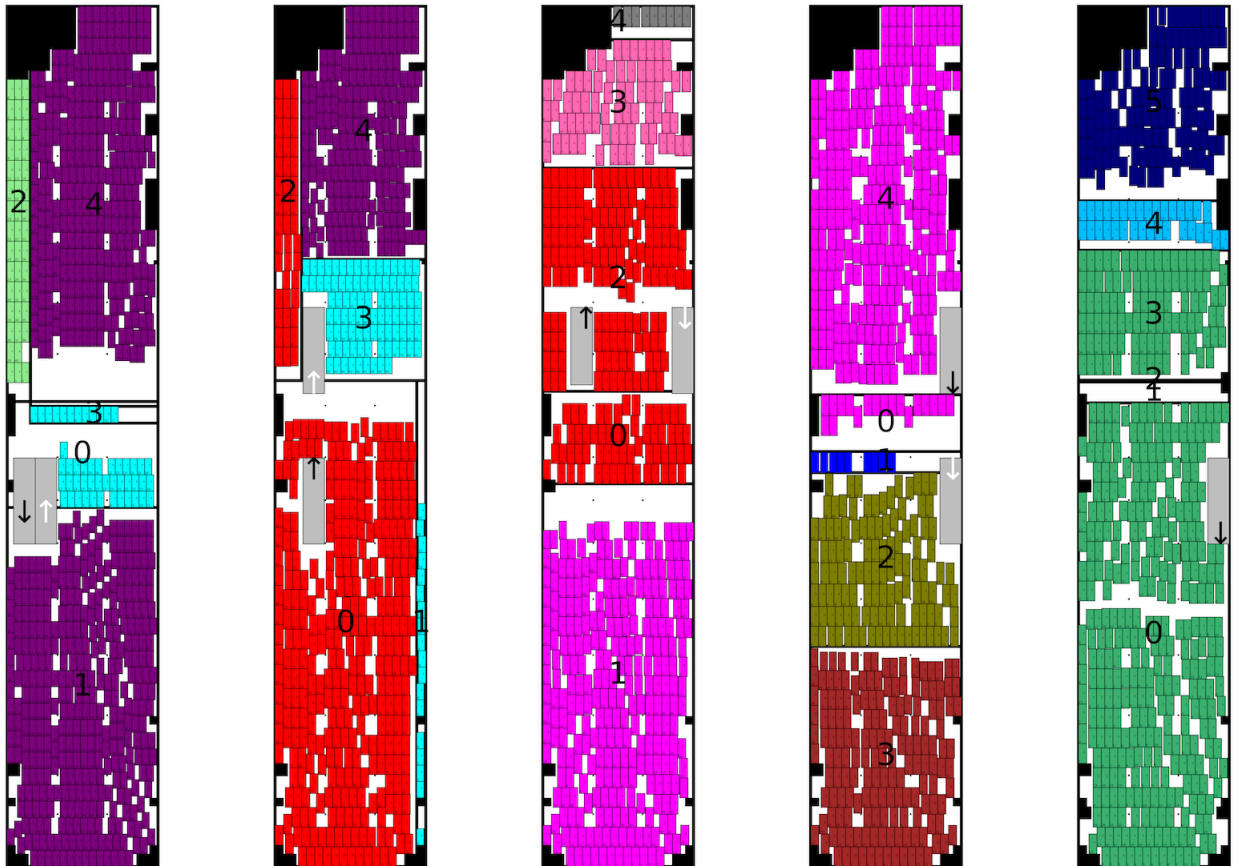


図 5.2: 第二段階の出力図の例

第6章 まとめと今後の研究計画

自動車運搬船への貨物積み付け計画における、車両配置計画に対して2段階の構築法を提案した。1段階目では、sequence-pair を用いた制約を追加し、soft-rectangle パッキング問題として定式化することで、搬入・搬出時の経路を確保した。2段階目では、レクトリニア図形を用いたパッキング問題として定式化し、駐車時の局所的なスペースを確保した。第一段階では整数計画ソルバーを用い、第二段階では bottom-left 法を用いて解を構築した。さらに以上の構築法で得られた解を初期解とし、近傍操作を行った結果、多くの問題例で解が改善され、より多くの車を詰め込むことができた。

今後の課題として、搬入搬出に関わる移動経路確保の正当性証明や、最終的な配置図を実際に業務で使われているものに寄せていくことが挙げられる。

謝辞

本研究の遂行にあたり、熱心な指導と助言を頂きました柳浦睦憲教授に深く感謝の意を表します。提案手法の検討やその有用性において活発に議論を頂きました、胡艶楠氏、竹田陽氏には大変お世話になりました。深くお礼申し上げます。日々の研究室生活においては柳浦研究室の皆様にお世話になりました。皆様のおかげで有意義な研究活動に勤しむことができました。深くお礼申し上げます。

参考文献

- [1] 「商船三井 プレスリリース」 <https://www.mol.co.jp/pr/2019/19072.html> (2022年2月10日)
- [2] 竹田陽, "自動車運搬船における貨物積載プランニングの席割問題に対する局所探索," 修士論文, 名古屋大学情報学研究科, 2022.
- [3] 今堀慎治, 梅谷俊治, "切り出し・詰込み問題とその応用 – (2) 長方形詰込み問題–," オペレーションズ・リサーチ, 50(2005), 335–340.
- [4] T. Ibaraki and K. Nakamura, "Packing Problems with Soft Rectangles," LNCS 4030, (2006), 13–27.
- [5] H. Murata, K. Fujiyoshi, S. Nakatake and Y. Kajitani, "VLSI Module Placement Based on Rectangle-Packing by the Sequence-Pair," IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems, 15(12), 1518–1524, December 1996.
- [6] R.C. Art, "An approach to the two dimensional irregular cutting stock problem," Ph.D. Thesis, Massachusetts Institute of Technology, 1966.
- [7] 今堀慎治, 胡艶楠, 橋本英樹, 柳浦睦憲, "Python による図形詰込みアルゴリズム入門," オペレーションズ・リサーチ, 63(2008), 762–769.
- [8] 柳浦睦憲, "局所探索法—反復改善に基づく最適化の基本戦略," オペレーションズ・リサーチ, 52(2007), 538–542.
- [9] J.O. Berkey, P.Y. Wang, "Two-Dimensional Finite Bin-Packing Algorithms," The Journal of the Operational Research Society, 38(1987), 423–429.